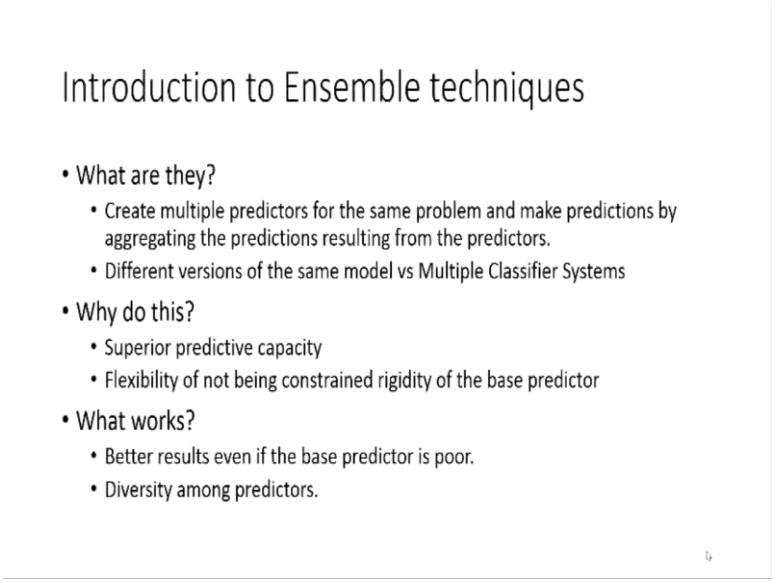
Introduction to Data Analytics
Prof. Nandan Sudarsanam
and Prof. B. Ravindran
Department of Management Studies
and Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 06
Lecture - 33
Ensemble Methods and Random Forests

Hello and welcome to our lecture on Ensemble Methods and Random Forests. So, ensemble methods are class of machine learning techniques and random forests are essentially is one such technique, it is an ensemble method and it is fairly popular one of that. So, we will be talking about ensemble method in general and we will also introduce random forests and the way we are going to do this is just to give an idea, how the algorithm works, what is the idea behind the algorithm and also give you some idea on where it works, why it works and so on. So, little bit conceptual also.

(Refer Slide Time: 01:01)



So, let us take a look at what essentially ensemble techniques are and the core idea behind on ensemble techniques is to create multiple predictors for the same problem. So, you are given a problem and here when I say problem I am talking about a prediction problem. So, it is a very much supervised learning by a given some input data with multiple inputs and you have an output and your job is to either to predict that output,

whether it is predicting a number or it is predicting a particular class like a classification task.

But, the core idea is that you given such a data set when you apply a particular methodology you create one predictor, whether it is regression or whether it is support vector machines or neural networks, whatever it is you create a single predictor; however, complex it is. But, with ensemble method you create multiple predictors for that exact same problem and you basically go about making predictions by aggregating the predictors.

There are couple of different ways of thinking about it, you can think about it as aggregating the predictors into this black box, which then you just call as one giant predictor or you can think of it as the model essentially is has multiple predictor. So, you use a data if created many predictors and tomorrow when somebody comes by and say can you make a prediction for me at this input point. You basically go and get the predictions from all the predictors and then, you take all these predictions and then you kind of aggregate them by taking an average or taking a vote or some method of aggregating those values.

The core idea behind how this works is that you invariably wind up injecting some kind of stochasticity into the data or the prediction algorithm. So, if you had a single data set and let us say the predictor of choice for you was decision trees or multiple regression, let just take those two examples. For a given data set, if you create a decision tree, you have a particular algorithm, so let us say you use the algorithm CART classification and regression trees.

So, we have discussed cart in previous lectures and so let say you go about creating a tree based on this data, you are going to wind up with a single tree, so it is a single predictor. How do you go up getting multiple predictors? Same thing with regression, you give it a data set and do a regression and you get a single regression equation, you do it again you are going to get the exact same regression equation. So, the core idea with ensemble methods are to kind of inject some form of stochasticity either in the data, that is use different data sets or we will talk about something called bagging shortly, but basically get some different versions of the data, let us just use the word versions, such that for each version you wind up with the different predictor.

So, if I give you one data set you will get one regression equation, if I give you another data set you will get another regression equation and that way you have different predictors. Another approach is to of course, inject that stochasticity into the algorithm, whether it is CART or neural networks or multiple regression there is a certain process which is what winds up giving you the predictors. So, you input the data into the process and outcomes the predictor.

Now, that process of creating the predictor itself if it had some randomness to it, then you might get different versions and we will talk about a couple of approaches that does that also. So, again even if you do not fully understand how to create different versions hang in there, because we will take a very concrete example with random forest and when we talk about there it will start to make a lot of sense. So, at this stage we still going to call you know keep it abstract, meaning you have a problem which means you have one grand data set and you basically instead of creating a single predictor to make predictions for this problem, you creating multiple predictors and hang in there in terms of how exactly we create multiple different predictors.

But, you create multiple predictors and then each predictor makes a prediction and then you aggregate those predictions to make your final prediction, so that is the core idea. Now, typically when the term ensemble method itself refers to the idea, where the core predictor, the base predictor is the same and what you are getting is just different versions. So, if the multiple predictors which you are creating are just slightly different regression equations, then technically that is ensemble method.

So, whereas if you had completely different predictors, like one is the regression equation and an another is a tree and another is a neural network, people tend to call this multiple classifiers systems, but over time I have actually see people just refer to all of them ensemble methods as well. So, the question is why do you want to do this and the answer to anything with supervised learning and prediction problems is, you do it if it gives you better predictive capacity and that is exactly what ensemble techniques do.

More specifically, they are especially able to take problems where the classifier or the predictor that you basically created is weak. So, the algorithm itself that you are using might be a little weak and you are able to actually make even with such weak predictors,

you are able to you know get a lot of them and you are able to completely get a much stronger over all prediction with this system.

Now, the reason and the potential reason that you get the superior predictive capacity is partly comes from the idea that any method is kind of constrained any fitting method. So, let us say if it is a regression equation or a decision tree is partly you know constrained by the rigidity of that predictor. So, for instance a multiple regression problem the rigidity could come from the mathematical equation itself being one of additive nature. So, it is like $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$ and so on.

So, the core idea being that it is a linear combination of these x's. Now, in the case of trees for instance you could have the problem where the rigidity comes from the tree like structure and so on. Now, when you see an ensemble method which is the essentially the aggregation of many such predictors, the overall structure has a lot more flexibility. So, you are no longer constraining two strong variables from falling in hierarchy or an order that is forcefully dictated by say a tree structure.

So, may be in one tree input variable x_1 plays a very dominant role and may be in another tree may x_2 plays a very dominant role and when you are aggregating these results you are able to form more complex behavioral interrelationship patterns between these input variables themselves. Now, it turns out that what really works with ensemble techniques is when there is a good amount of diversity among the predictors.

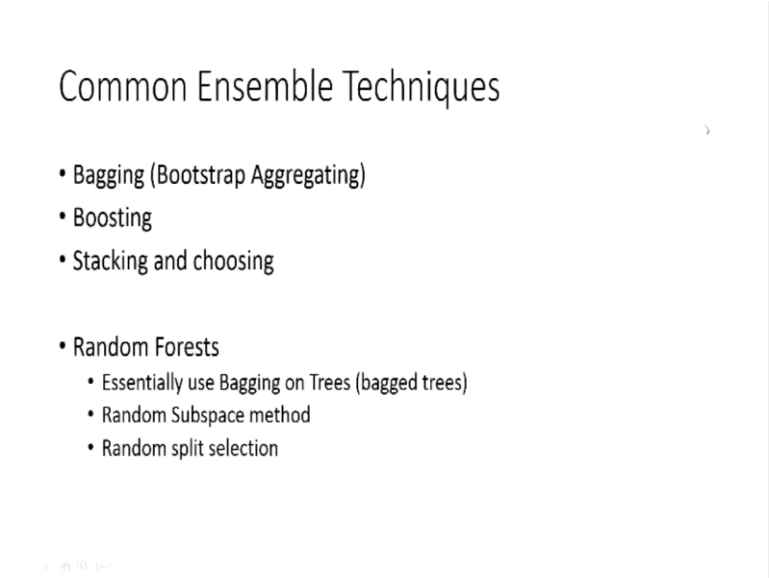
So, sometimes it is not so obvious that the state of the art in a particular technique is what world translate to the best ensemble method version of it. So, let me give you an example. So, let us say you had trees and even for a tree algorithm there are a lot of versions of constructing the tree, there are lot of metric based on which you can construct the tree we for instance had a problem and discussions on how you can use different metrics to build a tree like entropy or misclassification, you also had problems and trying to understand how you know what should be the minimum number of leaves at terminal nodes and what is the optimum level of pruning and so on.

So, it is not obvious that doing what is state of the art in a particular method will necessarily translate to the best version of on ensemble techniques. So, sometimes it turns out that even not splitting a tree in a very sensible fashion is fine and that works better when you are creating an ensemble of such trees because what tends to work with

these prediction trees is good amount of diversity among the trees and so that is something to definitely keep in mind that doing the state of the art when you are interested in creating a single predictor is not necessary the best thing to do when you are creating many versions of the same predictor that you want to combine you might opt for some the predictor that is poor or that is worse or that might be over fitting the data.

But, because you are now going to use it in the context of a random forest I mean context of an ensemble method, where you going to create many such versions of it and then add them together it might work just fine.

(Refer Slide Time: 11:24)



So, let me give you some concrete ideas on some ensemble techniques and we will start with the more basic ones. So, the first one is bagging or it is also called bootstrap aggregating, the idea here with bagging is that with bagging you are essentially doing all the work with the data. So, what you do is you are given a particular data set, so let say your data set has 200 points.

What you will do is, you will take those 200 points and put up in a big bucket, you will randomly sample a data point out and write it down and after that you will put this data point back into the bucket and then you will again randomly take another data point and write it down and like this and again put it back into the bucket and like this you will do this there are many version of it, but the most typical version is that you will do it 200

times, because were 200 data points start with. So, you take 200 random samples with replacement.

Now, what would happen if you did this is that you now have a set of 200 points and they are called 200 bootstrap points, which is not identical to the original 200 data points set. Why? Because, it is possible that by random chance I sometimes pick one particular point many times and it is possible that by random chance that a particular data point never got randomly picked all though I am picking 200.

But, because I am doing with the replacement there is a possibility of a double pick or triple pick and that it is possibility that I completely miss out the data. Now, when I do that I get a new data set of size 200 and then I go and I run my prediction algorithm on that be it whatever it is. Now, I do this same process many, many times, so I do this same process 20, 50, 100 times, where I create 20, 50 or 100 predictors by actually randomly sampling the data for the same number of data points that there are and creating bootstrap data set one that will create predictor one.

Go to the same thing again I will get bootstrap data set 2 and this time the random points random 200 points that I sampled want be identical to the 200 random points that I sampled before. So, now I have bootstrap data set 2 and that creates predictor 2 and so on I create 10, 20, 30, 100s of such predictors, 1000s of such predictors and I aggregate these predictors to get the results. Now, although I am side stepping for the second there this technique of bootstrapping and creating multiple predictors are actually goes beyond just ensemble methods and it has a lot of other interesting uses.

For instance, it is reasonably common to do this process to create multiple predictors to get an idea. So, out here what the reason we are doing this, so that we can create multiple versions, multiple predictors or you can think of it is multiple versions a type of predictor and to essentially make all these predictors predict when tomorrow prediction problem comes, tomorrow someone comes and says can you please predict me at this point I am going to ask each of these predictors to make a prediction and I am going to take their average or I am going to take some weighted average of them, but you can also use this to for instance get an idea of the uncertainty around your prediction.

So, a common practice is to use this in some way to get some kind of a confidence interval or a prediction interval and because you can still have the base data set, the one

that is not been bootstrap at all and create a predictor out of that and say that is my final prediction. But by the way am I 100 percent certain about my prediction or can I come up with some kind of a confidence bounds around my model or around my prediction that I make using my model and when you create many such predictors, you could use the many such predictors to predict at a given point to get a feel for the uncertainty that is there in my ability to predict a certain points.

So, I have an average guess based up off my algorithm, but can I built confidence bound using this bootstrapping technique. So, that is essentially the core idea behind bagging. We then come to boosting, the idea behind boosting is that again will typically dealing with weak classifiers or weak predictors and we are aggregating many of them. But, the way we would do that essentially is we will sequentially train each weak classifier, while this core approach in bagging is to create many different types of data sets and in some sense create many predictors that are essentially independent of each other.

Because, how well predict a one day has no barring on consequence on predictor two, because they are just working with different data sets and in some sense therefore, you can think of them as a parallel deployment, you create 200 different bootstrap data sets, you create 200 parallel you know parallely you can create 200 different predictors between boosting you are essentially looking at sequential process of doing the same thing and you would be creating many such predictors, many such predictors in a sequential way and these predictors themselves are weighted in some way that is usually related to the accuracy of each predictor.

But, the idea behind it being is sequential is the following, which is that you apply a weak predictor and you do the prediction process on the training data and you essentially weight the data and this is the most common version essentially of it, where the data is reweighted and the idea behind reweighting the data is to say oh I applied this predictor and it got all of these data point correct, oh and by the way it got the these other data points incorrect.

So, somewhere I want to weight the data points that it got incorrect. An incorrect in classification problem means, it did not classified correctly and incorrect in regression problem would mean that the square the deviation of the prediction from the actual is very high. So, essentially you kind of create this weighting structure on data points,

where points that are incorrectly predicted get a higher weight and so the next and that becomes the training data for the next iteration or the next predictor.

The next predictor will predict and then it gets a weight and it gets added to the previous predictors and you have at each stage n, you have n - 1 predictors that have done a prediction job and you have an idea of what those n - 1 predictors together are doing correctly or not. Because, this n - 1 predictors will be kind of each having a weight and those weights together will lead to a prediction process for the fixed data set.

Here unlike bagging you are not playing around with the data sets one fixed data set and you are essentially on that fixed data set you are applying these n - 1 predictors which are then getting weighted in some way and you have a final prediction and that prediction you are going to compare with training data to see which ones got tag correctly, which ones did not get tag correctly and then you are going to up the weight on the ones that did not get predicted correctly.

So, we are not changing the data points themselves we are changing the weights or the importance that you are giving to some data points versus the others. So, think of this as the prediction problem, where you are actually going and saying oh these data points are more important, which means you better as a predictor you better get those correct, oh and by the way here are these other data points that I think are less important. So, in some way by dynamically doing this you are really kind of making sure that some versions are actually classifying some areas correctly; the areas that tend to get misclassified are not tending to get misclassified.

So, that is the core idea behind boosting. In addition to this you have various other approaches, the core for instance the core approach of stacking is one, where you actually have completely different predictors. So, you take the data set and then you apply like a neural networks and then you might apply you know something else you know support vector machines whatever. So, you apply like completely different methods or you could have related methods, but then again the core idea is that with stacking you have another algorithm, another prediction algorithm which decides how much weight to give to each of these sub predictors.

And therefore, it is essentially like a stacking many different predictors using another prediction algorithm with the core idea of choosing you are doing something different

you are pretty much allowing for it is often called as like a break off context, because all you are doing is you are taking a whole battery of possible methods KNN, support vector machines, neural networks, decision trees, regressions, apply each of them to the data set use something like cross validation, which is the topic we have discussed before and to determine which one works best and use that in some way or use that to give weights to the methods themselves.

So, while stacking involves using many predictors and then using some kind of a combiner, you can call that the final thing like some kind of a combiner to combine the predictions of all these algorithms then and choosing is more of instead of thinking of it is combining it is more of trying different things and using cross, cross validation. So, it is like you have a bucket of models and then you try cross validation across them and try to pick the ones that perform really well.

So, this should give you a rough idea of the common ensemble techniques, I am going to talk about one specific one called random forests and these are fairly popular set of supervised learning techniques, random forests essentially use decision trees. So, they use the basic tree algorithms and they are essentially an application of ensemble techniques using the base what have been using calling the predictor, here is specifically going to be a tree predictor, the basic idea here with the random forest is that they use bagging bootstrap aggregating on trees.

So, use the core concept of sampling with replacement you create many data sets and on each data set you will apply your tree algorithm may be it is a CART and once it is done you typically do not do things like pruning the trees you just let the tree fully kind of grow out and you create many such trees and when you do that, that is kind of just first step and these are actually called bagged trees.

And the idea would be that tomorrow when somebody comes and asks you to make a prediction you do not just drop that input data point down one tree you drop it out on all these trees and so each tree will come up with some prediction and the prediction could be a classification or a regression and you aggregate the results, either take a average or you kind of have them both and then you come up with final answer.

Now, in addition now just doing this process is often just called bagged trees, but if you essentially take bagged trees and you injects stochasticity through other means into the

algorithm itself. So, what we mean is... you have done bagging of trees, but one problem that they used to find with even this approach is that there could be like this one very prominent variable which always chooses to get split first and that split is not allowing you to see certain patterns in other variables.

And even if you can do bootstrap aggregating, you are still getting more or less same set of data points, you are getting very similar looking trees. So, one approach there is called the random sub space method and the idea here is that at each split you present the algorithm with a sub set of the features on which it can be split. So, at each split in addition to the fact that each tree is built on a bootstrap data set at each split you do not provide the algorithm, which is choosing to make a particular split all the features that it can potentially split on.

So, you give it a sub set of the features and that is why it is called random sub space. So, it is a randomly selected sub set that is given at each step and based on that subset go ahead create a split and then at the next state I will give another subset. Now, that is not necessarily the only way to do it there are other approaches in addition to that there is the random split selection for instance,

So, when the random split selection is applied just on top of bagged trees, the way it would work is that I would randomly select a point in each feature and then I would give you that the set of features and you choose the one that you like the most meaning. So, I will say on variable a the location you can split is not something that I am giving you a choice about, but you can choose which variable to split on. So, I am going to randomly select the point and I believe these are called extremely randomized trees also the extra trees.

But, the idea here again is that you will fix, you will randomly select a point on that feature on which the split can occur and like that for every feature you will select the point and you present the algorithm with these 20 features and say pick the feature that you want to make the split on in order to kind of become a good predictor or in order to maximize your criteria.

Another thing is at each step with random split selection variant of that approach is to say at each step you can only randomly split, you can only split on a subset of we will pick for you the best five splits that can do based on your algorithm, but I am going to

randomly choose one of those five. So, essentially if you had the algorithm kind of say there are 20 features and by the way on those 20 features if I was applying cart I would split on feature x at point y instead you say tell me the top five features that you want to split on and you choose the point where you want to split.

But, tell me the top five or top three features you want to split on and once you give me the top n I am going to randomly choose one from that and then you split on that. So, that is another variant of this notion randomly splitting. So, the core idea here that you should see is that both random subspace methods and random split selections or extra trees all focus on the idea of injecting scarcity more or less in the algorithm that is going about doing this split.

Whereas, the core notion of bagging trees works on the principle of giving different data sets to the problem and the randomness comes from the data set itself, but when like a one of these algorithm with bagged trees. So, like random subspace or random split selection you are essentially applying a random forest analysis. So, essentially when you have a stochasticity that is being applied on the data itself through bagging, you are creating bagged trees and when you have scarcity in the algorithm itself you can do it through like random subspace or random split selection and when these two are used together essentially when you are doing random forest, you are applying the procedure of random forests to create this predictor. So, I hope that gives you an idea of ensemble techniques and more specifically on how random forest work and we look forward to seeing you in the next lecture.

Thank you.